VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Absolvování individuální odborné praxe

Individual Professional Practice in the Company

2018 Petr Dolejší

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Zadání bakalářské práce

Student: **Petr Dolejší**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2612R025 Informatika a výpočetní technika

Téma: Absolvování individuální odborné praxe
Individual Professional Practice in the Company

Jazyk vypracování: čeština

Zásady pro vypracování:

1. Student vykoná individuální praxi ve firmě: SEKO system s.r.o.
2. Struktura závěrečné zprávy:
   a) Popis odborného zaměření firmy, u které student vykonal odbornou praxi a popis pracovního zařazení studenta.
   b) Seznam úkolů zadaných studentovi v průběhu odborné praxe s vyjádřením jejich časové náročnosti.
   c) Zvolený postup řešení zadaných úkolů.
   d) Teoretické a praktické znalosti a dovednosti získané v průběhu studia uplatněné studentem v průběhu odborné praxe.
   e) Znalosti či dovednosti scházející studentovi v průběhu odborné praxe.
   f) Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení.

Seznam doporučené odborné literatury:

Podle pokynů konzultanta, který vede odbornou praxi studenta.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **doc. Dr. Ing. Eduard Sojka**

Konzultant bakalářské práce: Ing. Jan Chobot

Datum zadání: 01.09.2017
Datum odevzdání: 30.04.2018

doc. Ing. Jan Platoš, Ph.D.
vedoucí katedry

prof. Ing. Pavel Brandštetter, CSc.
děkan fakulty

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 23. dubna 2018 .....

Souhlasím se zveřejněním této bakalářské práce dle požadavků čl. 26, odst. 9 Studijního a zkušebního řádu pro studium v bakalářských programech VŠB-TU Ostrava.

V Ostravě 23. dubna 2018

SEKO system s.r.o.
Václavská 2027/11, 709 00 **Ostrava**
DIČ: CZ25848623
tel/fax: 596 619 885

Rád bych na tomto místě poděkoval mé rodině za podporu při studiu i firmě SEKO system s.r.o. za možnost vykonat odbornou praxi.

Abstrakt

Práce popisuje průběh vykonávání bakalářské práce formou individuální odborné praxe ve firmě SEKO system s.r.o. V průběhu praxe jsem plnil úkoly a vystupoval jsem v roli vývojáře. Úkol byl vytvořit systém pro řízení servisního oddělení.

Klíčová slova: Bakalářská praxe, C#, ASP.NET MVC

Abstract

This bachelor thesis describes the course of the individual professional practise in the SEKO system s.r.o. company. During my practice I have been fulfilling tasks and I worked as a developer. The task was to create a service management system.

Key Words: Bachelor practice, C#, ASP.NET MVC

Obsah

Seznam použitých zkratek a symbolů

ABS	–	Vyvíjený software v rámci odborné praxe
AD	–	Active Directory
ASP.NET	–	Framework pro vytváření webů
CSS	–	Cascading Style Sheets
HTML	–	HyperText Markup Language
IIS	–	Internet Information Services
IS	–	Informační systém
MS	–	Microsoft
MS SQL	–	Microsoft SQL Server
MVC	–	Model - View - Controller
např.	–	například
PDF	–	Portable Document Format
SQL	–	Structured Query Language
tzv.	–	takzvaný
XLS	–	Soubor programu MS Office Excel

Seznam obrázků

Seznam výpisů zdrojového kódu

1 Úvod

Bakalářskou práci jsem vykonával ve společnosti SEKO system s.r.o. sídlící v Ostravě. Firma se zabývá prodejem a servisem kopírovací a prezentační techniky. Pro své účely používají svůj vlastní systém řízení servisu - ABS. Současná aplikace je optimalizován pro OS Windows XP, proto v dnešní době, kdy se využívají již Windows 10, je práce s tímto systémem velmi náročná.

Firma měla pouze jeden požadavek na zvolené technologie. Celý systém má být jako webová stránka - z důvodu multiplatformnosti (ve firmě se používají pro stolní počítače Windows, macOS a na mobilních zařízení Android a iOS). Rozhodl jsem se použít, na základě stávající technologické vybavenosti firmy C# ASP.NET MVC.

Postupně jsem byl seznámen se starým programem, spolupracoval jsem s více lidmi, kteří tento systém používají. Poté mi bylo vysvětleno, co nový systém má dělat a v čem se má lišit oproti staré aplikaci. Již na začátku vykonávání praxe bylo jasné, že v průběhu praxe nelze stihnout všechny požadavky firmy, proto jsem s firmou domluven, že na projektu budu pokračovat i mimo vykonávání odborné praxe.

Protože firma nemá svůj vývojářský tým, na celém projektu pracuji sám. Měl jsem ale možnost konzultovat projekt s mým známým, kterého firma schválila, který jakožto bývalý student stejného oboru, jako já a následně programátor ve firmě TIETO, má k programování v C# i ASP.NET MVC blízko.

Všechny výpisy zdrojového kódu v této dokumentaci jsou ukázky z aplikace a tudíž nejsou vymyšleny.

2 Zadaný úkol

IS má nejprve obsahovat řízení servisního oddělení, s možností do budoucna, dodělat i řízení obchodního oddělení. V této práci budu řešit jen servisní část.

Systém má obsahovat evidenci strojů (kopírovací stroj/projektor/něco jiného), evidenci smluv a evidenci oprav. Stará verze řeší pouze stoje a opravy. Smlouvy se řeší jiným systémem, ale navzájem nejdou propojit. Cíl bylo, aby uživatel měl co nejméně možností k vypsání textových polí a spíše měl možnosti k výběru. Zároveň bylo vyžadováno, aby se důležitá data nepřepisovala, ale aby byla vedena historie. V systému zatím bude 6 uživatelských rolí, do budoucna se rozšíří na další.

Většina kopírovacích strojů dokáže posílat mailem v pravidelně nastavených intervalech počítadla. Pro tyto počítadla jsem měl za úkol vytvořit Windows službu pro zpracování těchto mailů.

2.1 Evidence strojů a příslušenství

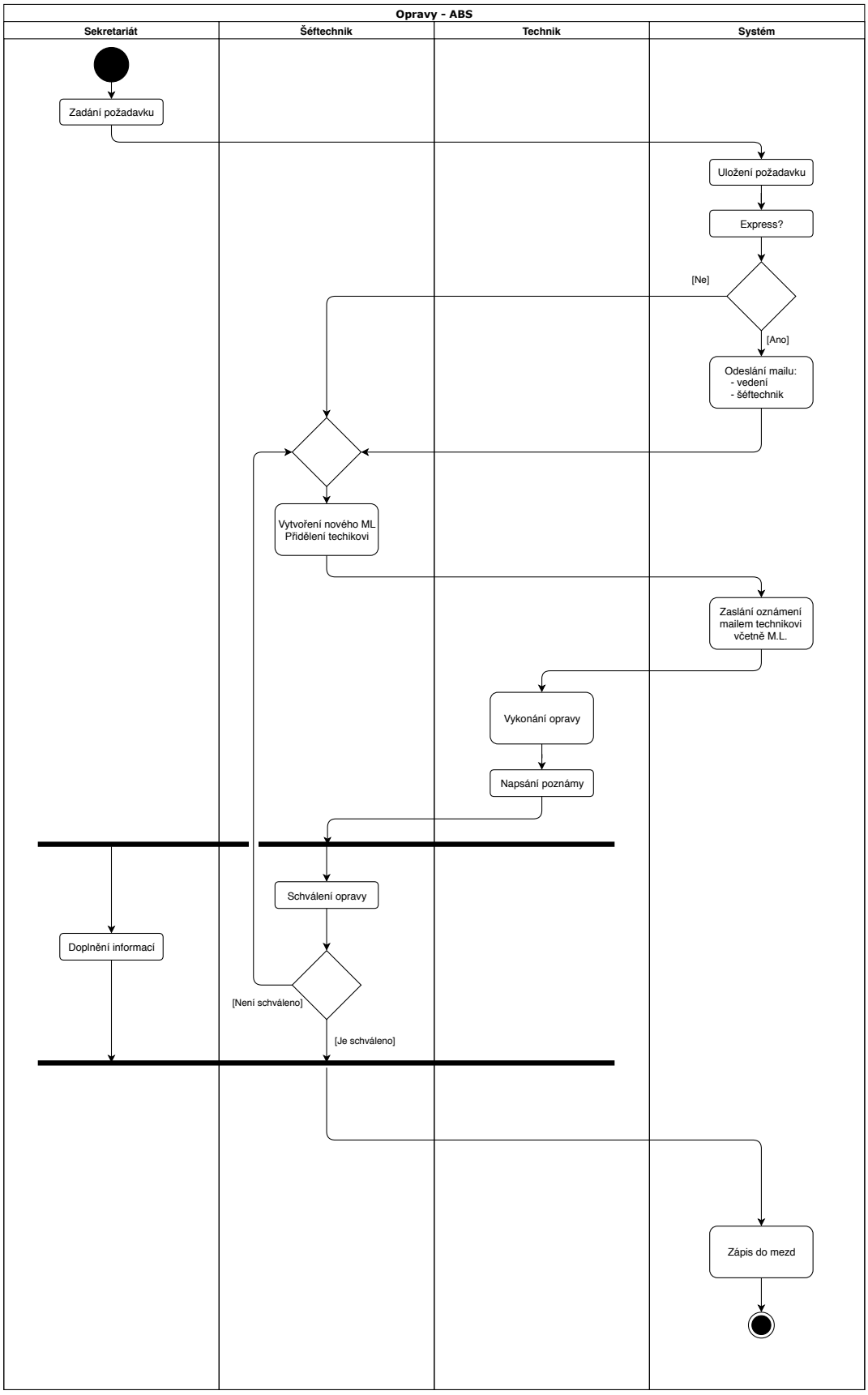
Nový systém se měl dívat na stroj, jako na samostatnou jednotku, která lze někam přiřadit a pak následně odebrat. To je rozdíl, oproti starému systému, kde stoj lze pouze přiřadit a pro následné přiřazení, například jiné smlouvě se musí stroj znovu vytvořit. To samé platí i pro práci s příslušenstvím. V rámci tonerů, které se používají, je vyžadováno, aby se hlídalo používání tonerů. Proto šablona pro toner obsahuje údaj s orientačním počtem výtisků na jeden toner. Díky toho se bude moci hlídat, když zákazník bude chtít nový toner, zda někde může nějaký mít schovaný a nebo ho opravdu potřebuje.

2.2 Evidence smluv

V rámci evidence smluv je kladen důraz na vkládání. Byl vytvořený celý proces vkládání, který obsahuje části, jako vytvoření smlouvy, vložení údajů, stav čekání na podpis, možnost vygenerovat PDF soubor k vytištění a podpisu. Generovaný PDF soubor má obsahovat nejen zadané informace, ale i obchodní podmínky, které se mají řešit v samostatné části. V některých případech se vkládají i podmínky pro software. V rámci smluv se mají řešit i dodatky, kdy se může na smlouvě cokoliv změnit za předpokladu, že tato úprava bude dohledatelná.

2.3 Evidence oprav

Pro opravy se nastavil nový postup, oproti starému systému. Pro každou opravu se bude moci vytvářet několik montážních listů - PDF soubor, který obsahuje vyplněnou hlavičku a další důležité informace. S tím jezdí technik po opravách a zákazník ho podepisuje. Po vyplnění poznámky technikem, sekretářka zapíše hodnoty zapíše do systému, na základě kterých se bude počítat mzda.



Obrázek 1: Aktivitní diagram opravy

3 Windows služba

Většina kopírovacích strojů je schopna posílat automaticky počítadla mailem (někdy to nejde, protože to místní síť nedovoluje). V tuto chvíli sekretářka tyto mail ručně zpracovává. Za úkol bylo vypracovat Windows službu, která poběží na serveru. Služba se bude připojovat na Emailovou schránku a bude zpracovávat počítadla a vkládat vyhledané údaje do databáze.

```
[Model Name],Dev-224e-sekretariat
[Serial Number], A5C4121
[Send Date],26/03/18
[Total Counter],0013429
[Total Color Counter],0001166
[Total Black Counter],0012263
[Total Scan/Fax Counter],0002462
```

Výpis 1: Počítadlo - část mailu (upraveno)

4 Hlavní část - vytvoření software ABS

4.1 Použité technologie a nástroje

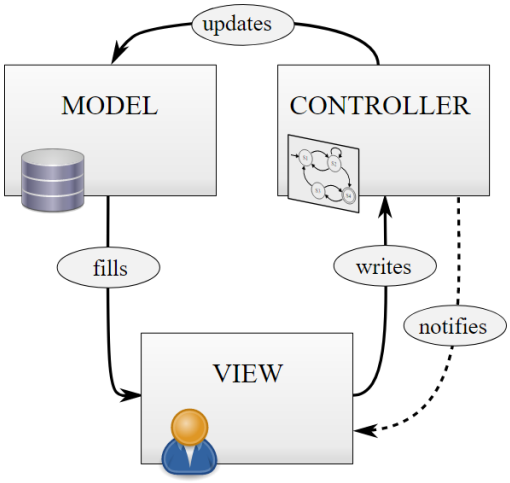
- MS Visual Studio - vývojové prostředí pro vývoj aplikací postavených na technologii od společnosti Microsoft
- MS SQL Server Management Studio - nástroj pro správu databází na MS SQL
- SQL Operations Studio - nový nástroj Microsoft pro správu databází, není tak těžkopádný, jako MS SSMS - zatím jen verze Preview
- MS SQL Server - databázový server od Microsoftu
- ASP.NET MVC - framework pro vytváření dynamických internetových aplikací

4.1.1 ASP.NET MVC

Ve firmě se používá Windows Server s nainstalovaným serverem IIS a na něm již běží ASP.NET webová aplikace, proto jsem zvolil tuto cestu. Při volbě frameworku jsem na základě doporučení zvolil ASP.NET MVC.

Tento framework (ASP.NET MVC) jsem před začátkem práce na této bakalářské práci jsem neznal, proto jsem nejprve začal studiem. Využíval jsem tutoriály, na kterých jsem se s tímto frameworkem naučil pracovat [2].

Návrhový vzor MVC obsahuje tři komponenty a to: Model, View, Controller. Cílem je oddělit logiku od výstupu.



Obrázek 2: Diagram MVC

4.1.1.1 Model

Model obsahuje logiku a vše, co do ní spadá. Model neví, odkud data přicházejí, ani, kam půjdou.

```
[DisplayName("Typ smlouvy")]
[Required(ErrorMessage = "Typ smlouvy je vyžadován")]
public virtual int Type { get; set; }

[DisplayName("Ico firmy")]
[Required(ErrorMessage = "Ičo firmy je vyžadováno")]
[StringLength(20, ErrorMessage = "Maximálně je povoleno {1} znaků")]
public virtual string IcoFirmy { get; set; }

[DisplayName("Kod firmy")]
[Required(ErrorMessage = "Kód firmy je vyžadován")]
[StringLength(20, ErrorMessage = "Maximálně je povoleno {1} znaků")]
public virtual string KodFirmy { get; set; }
```

Výpis 2: Model - část kódu modelu smlouvy

4.1.1.2 View

View má na starosti zobrazení výstupu uživateli. Základ je HTML kód a může obsahovat Razor kód - slouží pro vkládání C# kódu do HTML.

Tato část by neměla obsahovat žádné výpočty. Kontroler může předávat data View několika způsoby. Jako hlavní způsob je model stránky. Může být jen jeden, ale například pro nějaký výpis se hodí IList modelů. Další předávání je dynamicky alokovaný ViewBag nebo TempData.

```
<div class="pull-right box-tools">
    @{
        if (Model.Locked == 0) {
            using (Html.BeginForm("Delete", "Contracts", FormMethod.Post)) {
                @Html.ActionLink("Upravit", "Edit", "Contracts", new {id = Model
                    .ID}, new {@class = "btn btn-success btn-xs"})
                <input type="text" name="id" value="@Model.ID" hidden/>
                <button type="submit" class="btn btn-danger btn-xs" onclick="
                    return confirm('Přejete si opravdu smazat @Model.ID?')">
                    Smazat</button>
            }
        }
    }
</div>
```

Výpis 3: View - část výpisu smlouvy

4.1.1.3 Controller Prostředník, se kterým komunikuje uživatel, model a view. Tyto komponenty také propojuje.

```
public ActionResult AddType (string id)
{
    ViewBag.ID = id;
    return View(new TemplateTypeAbsDao().GetAll());
}

[HttpPost]
public ActionResult AddType (string idContract, int idType)
{
    try
    {
        var item = new ContractTypeAbs()
        {
            Type = new TemplateTypeAbs() { ID = idType },
            Contract = new ContractAbs() { ID = idContract }
        };
        new ContractTypeAbsDao().Create(item);
        TempData["message-success"] = "Typ byl přidán.";
        return RedirectToAction("Detail", "Contracts", new { id = idContract })
            ;
    }
    catch (Exception)
    {
        TempData["message-error"] = "Vyskytla se chyba. Typ nebyl přidán.";
    }
    return RedirectToAction("Detail", "Contracts", new { id = idContract });
}
```

Výpis 4: Controller - část kontroléru smluv

Jak lze vidět v předchozí ukázce, obě akce se stejně jmenují. Horní akce se provede po zavolání uživatelem a spodní až po odeslání formuláře pomocí POST.

4.1.1.4 RouteConfig

RouteConfig je zodpovědný za správné vybrání kontroleru a akce z adresy, která je vyvolaná uživatelem. V rámci konfigurace se volí výchozí kontroler (Index) a akce pro případ, že adresa žádný neobsahuje. V mém případě nebyla potřeba RouteConfig upravovat, stačil mi výchozí vygenerovaný soubor.

Hlavní důvod, proč se tento upravuje je, aby vývojář mohl vytvořit "hezké odkazy"- vhodné např. pro SEO. Další možností je, že pro parametry se mohou nastavit pravidla, například regulární výraz.

4.1.2 NHibernate

NHibernate je objektově-relační mapper. Je to prostředník mezi aplikací a databází. Při změně databázového systému stačí jen upravit jeho konfiguraci. NHibernate generuje SQL dotazy tak, aby byly nejefektivnější a díky toho se vývojář může více soustředit na aplikaci. Před použitím tohoto v frameworku v této práci, jsem ho neznal a nikdy nepoužil. Důvod, proč jsem ho byl, že byl použit v již zmíněných tutoriálech.

Důvod, proč je výhodné používat tyto frameworky (např. NHibernate, Entity Framework) je, že po konfiguraci samotného frameworku je mapování každé tabulky velmi snadné. V případě použitého NHibernate, stačí jeden XML soubor pro každou tabulku. Ten obsahuje informace o použitém modelu a tabulky v databázi. Poté každý řádek třídy obsahuje parametr použité property v modelu a atributu v tabulce. V NHibernate využívám 4 typy mapování jednotlivých atributů:

- ID - použití pro atribut s Identity (generování unikátních ID v databázi)
- CompositeID - pro složené primární klíče v databázi
- Property - klasické mapování atributu bez žádného vztahu
- Many to one - jednoduché mapování atributu se vztahem 1:N

U mapování se nezapisují datové typy, proto je mapování velmi rychlé a jednoduché. Při spuštění aplikace se NHibernate zkontroluje, zda datové typy v modelu se shodují v databázi a případně zahlásí chybu.

```
<class name="AgreementAbs" table="Agreements" lazy="true">
    <id name="ID" column="ID" />
    <property name="KlientNumber" column="KlientNumber" />
    <property name="Location" column="Location" />
    <property name="Type" column="Type"/>
```

Výpis 5: Část mapování tabulky v NHibernate

Po namapování tabulek stačí již skládat jednotlivé dotazy. Dotaz se může být buď z klasického SQL, skládání z metod (QueryOver) a nebo využití LINQu.

4.2 Prerekvizity

Protože systém dělám sám, musel jsem udělat i jiné úkony, než jen programování. Od nastavení IIS, přes hledání vhodných knihoven (např. převod Excelu do PDF, spojování PDF souborů do jednoho, atd.), komunikace se správcem serveru, atd. Pro design jsem našel šablonu, kterou mi firma schválila, založenou na Bootstrapu, pro optimalizaci na mobilní telefony i počítače zároveň. Měl jsem možnost pracovat se starým ABS.

Pro lepší představu, co systém má dělat, měl jsem možnost jet do Prostějova na předváděcí akci systému, společnosti BüroKomplet. Tento systém je podobný tomu, co v této odborné praxi řeším. Já jsem si ujasnil, co se ode mě očekává a vedení firmy, se kterým jsem na tuto akci jel, si ujasnilo některé věci, které chtějí od nového ABS.

V rámci firmy jsem měl konzultace na interaktivní tabuli se zaměstnanci, tedy budoucími uživateli, kde jsme společně řešili různé věci tak, aby splňovaly procesy, které si vymyslelo vedení a zároveň, aby se se systémem uživatelům dobře pracovalo.

4.3 Naprogramování Windows služby

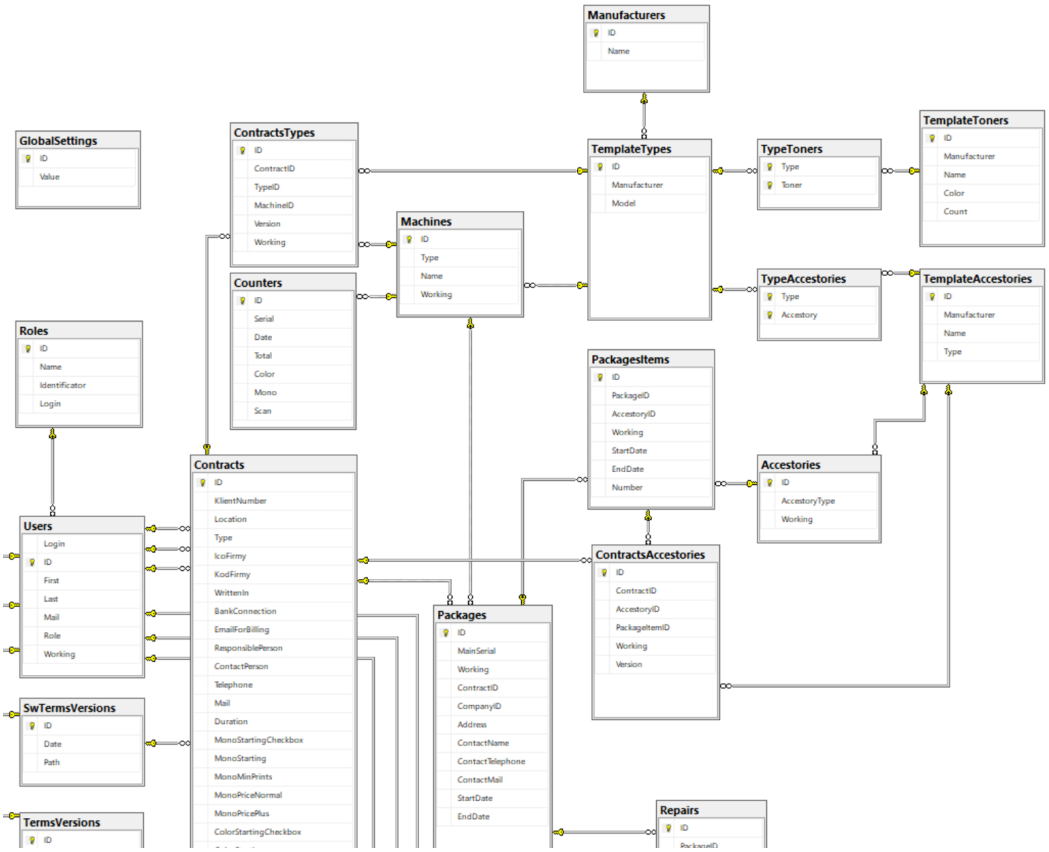
Služba se připojuje na mail pomocí knihovny S22.Imap [1], která podporuje IDLE připojení na schránku - služba vyčkává na notifikace od serveru, nezatěžuje server ani připojení na pravidelné dotazy na poštovní server, zda není nová zpráva.

Po oznámení, že přišla nová zpráva, služba pomocí třídy Regex text zpracuje a poté je vloží do databáze. V případě úspěchu zprávu přesune do složky se zpracovanými počítadly. Nezpracované se vloží do složky nezpracované a v systému bude rozhraní na prohlížení těchto mailů. V případě, že nastane chyba, například výpadek internetu, služba obsahuje tzv. půlnoční čtení, kdy se prohledá celá složka doručených mailů a přečtou se tak zprávy, které se nepřečetly během výpadku.

Služba je hotová, již několik měsíců čte data z počítadel. Postupem času již jenom opravuji chyby (služba vypisuje log), které se někdy vyskytnou.

4.4 Vytvoření databáze

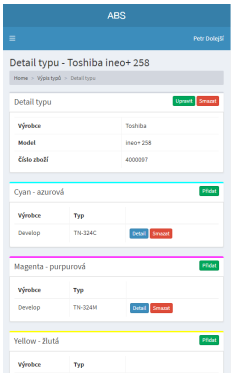
Pro systém jsem navrhl databázi. Jako databázový server se používá MS SQL server. Pro přístup k serveru jsem hlavně využíval MS SQL Server Management Studio. V databázi využívám 24 tabulek, které jsou většinou ve vztahu 1:N. Díky relacím (využití cizích klíčů), které jsou v databázi vytvořené, NHibernate tyto vztahy rozpozná a pro programátora vytvoří jednotlivé napojení, takže využívá jen třídy, bez žádného mapování.



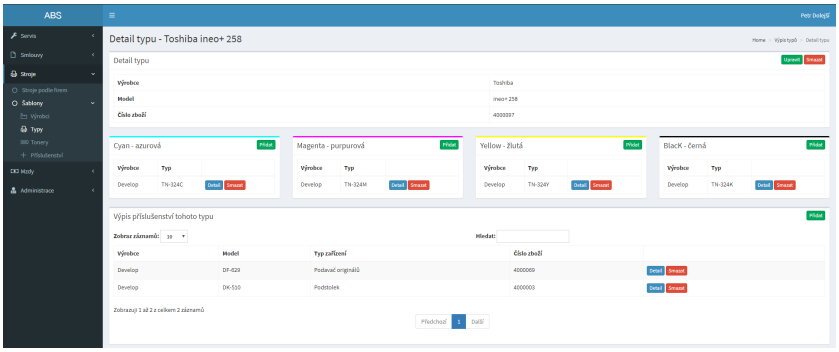
Obrázek 3: Náhled části databázových tabulek z projektu

4.5 Návrh GUI

Pro uživatelské rozhraní jsem použil šablonu AdminLTE [8] založenou na frameworku Bootstrap. Tento framework obsahuje stylování různých komponent a možnost web naprogramovat tak, aby byl responzivní. Díky toho se aplikace zobrazí jinak na mobilním telefonu, tabletu, menším monitoru a na velkém monitoru. Několik věcí v stylování jsem si upravil.



(a) Na telefonu



(b) V počítači

Obrázek 4: Ukázka GUI

Při používání grafických možností šablony jsem kladl důraz na použití barev a dalších prvků, aby uživatel měl jednodušší práci. Například: červená tlačítka - smazání, odebrání; zelená tlačítka - přidat, aktivovat, upravit; modré tlačítko - zobrazit detail, soubor. Jedna z mých úprav stylování, například na obrázcích lze nepatrně vidět horní linka boxů, značí barvu toneru, se kterým je box spojen, aniž by to rušilo, přestože jde o pestré barvy.

Díky použití frameworku vypadají všechny prvky podobně, pro uživatel je jednodušší se systémem naučit a pracovat s ním.

4.6 Přihlašování

Každý zaměstnanec má svůj účet v Active Directory, se kterým se přihlašuje do svého PC. Proto jsem této skutečnosti využil a proto se do nového ABS uživatelé přihlašují svým AD uživatelským jménem a heslem. Uživatelské role již řeším mimo AD, protože do AD nemám správcovský přístup (spravuje to externí firma). Pro toto jsem udělal administrační rozhraní sám.

Pro řešení přihlašování v aplikaci využívám třídu MembershipProvider a pro správu uživatelských rolí třídu RoleProvider. Výhoda implementace třídy RoleProvider je, že pak lze jednoduše přidělovat kontrolerům nebo jednotlivým akcím v kontrolerech role, pro které jsou dostupné. To samé lze využít přímo ve View.

Tato část je celá hotová a funguje bez problémů.

4.7 Napojení na účetnictví

Aby v novém systému nemusela být řešena další evidence firem, využil jsem možnosti se připojit do účetnictví a tam číst v tabulce firem. Jedná se o účetnictví Klient firmy SoftApp. Proto stačí vytvořit firmu v účetnictví - primární systém na vedení těchto záznamů - a díky toho je i v ABS. V této databázi (tabulce) využívám jen čtení.

V databázi jsou uloženy informace jako kód, IČO, DIČ, název, adresa a kontaktní osoba. Informační systém je proto přizpůsoben pro logiku tohoto účetnictví.

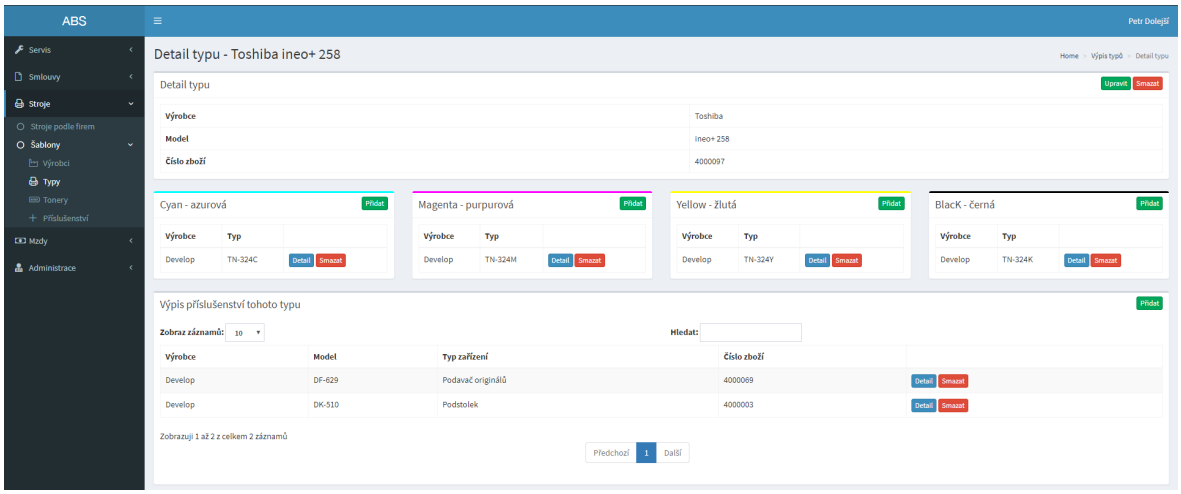
Pro tento úkol jsem se přihlásil do databáze od účetnictví a na její základě jsem vytvořil potřebné mapování. Mapování i komunikace s databází je zcela funkční.

4.8 Správa šablon

Pro zmenšení množství vyplňování textovými poli ve formulářích uživatelem, jsou v IS vytvořeny tzv. šablony, kdy se vytvoří výrobce, model, příslušenství a tonery. U tonerů se zaznamenává počet výtisků pro další kontrolu.

Některé šablony lze propojit a přiřadit navzájem, například příslušenství lze přiřadit k typům strojů, proto se nemůže stát, že se přiřadí nekompatibilní zařízení navzájem.

Šablony jsou již vytvořené a tedy hotové.



Obrázek 5: Šablona pro stroj s výpisem propojených tonerů a příslušenství

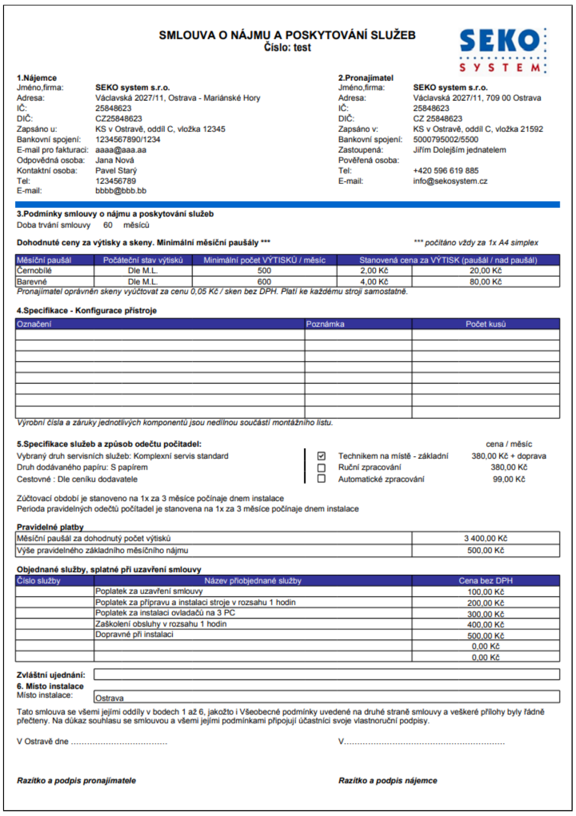
4.9 Export do PDF

Veškeré souborové výstupy, by na přání firmy měly být exportovány do PDF. Stávající formuláře, které vyplňují lidé, jsou dělané v Excelu. Proto jsem hledal vhodnou knihovnu pro vkládání dat do buněk listu v XLS souboru a pak následný export do PDF. K této operaci využívám knihovnu GemBox.Spreadsheet [4].

V rámci smluv se do již vyexportovaného souboru PDF vkládají na konec souboru všeobecné obchodní podmínky, případně podle potřeby, podmínky pro software pomocí knihovny PDFsharp [3]. I ostatní dokumenty se exportují do PDF.

Montážní listy pro budoucí mobilní aplikaci obsahují vygenerovaný QR kód, který generuji pomocí knihovny QRcoder [5].

Všechny potřebné soubory se již exportují tak, jak mají.



SMLOUVA O NÁJMU A POSKYTOVÁNÍ SLUŽEB
Číslo: test

1.Nájemce
Jméno,firma: **SEKO system s.r.o.**
Adresa: Václavská 2027/11, Ostrava - Mariánské Hory
IČ: 25848623
DIČ: CZ25848623
Zapsáno u: KS v Ostravě, oddíl C, vložka 12345
Bankovní spojení: 1234567890/1234
E-mail pro fakturaci: aaaa@aaa.aa
Odpovědná osoba: Jana Nová
Kontaktní osoba: Pavel Starý
Tel: 123456789
E-mail: bbbb@bbb.bb

2.Pronajímatel
Jméno,firma: **SEKO system s.r.o.**
Adresa: Václavská 2027/11, 709 00 Ostrava
IČ: 25848623
DIČ: CZ 25848623
Zapsáno v: KS v Ostravě, oddíl C, vložka 21592
Bankovní spojení: 5000795002/5500
Zastoupená: Jiřím Dolejším jednatelem
Pověřená osoba:
Tel: +420 596 619 885
E-mail: info@sekosystem.cz

3.Podmínky smlouvy o nájmu a poskytování služeb
Doba trvání smlouvy 60 měsíců

Dohodnuté ceny za výtisky a skeny. Minimální měsíční paušály *** **** počítáno vždy za 1x A4 simplex*

Měsíční paušál	Počáteční stav výtisků	Minimální počet VÝTISKŮ / měsíc	Stanovená cena za VÝTISK (paušál / nad paušál)	
Černobílé	Dle M.L.	500	2,00 Kč	20,00 Kč
Barevné	Dle M.L.	600	4,00 Kč	80,00 Kč

Pronajímatel oprávněn skeny vyúčtovat za cenu 0,05 Kč / sken bez DPH. Platí ke každému stroji samostatně.

4.Specifikace - Konfigurace přístroje

Označení	Poznámka	Počet kusů

Výrobní čísla a záruky jednotlivých komponentů jsou nedílnou součástí montážního listu.

5.Specifikace služeb a způsob odečtu počítadel:
Vybraný druh servisních služeb: Komplexní servis standard
Druh dodávaného papíru: S papírem
Cestovné : Dle ceníku dodavatele

		cena / měsíc
☑	Technikem na místě - základní	380,00 Kč + doprava
☐	Ruční zpracování	380,00 Kč
☐	Automatické zpracování	99,00 Kč

Zúčtovací období je stanoveno na 1x za 3 měsíce počínaje dnem instalace
Perioda pravidelných odečtů počítadel je stanovena na 1x za 3 měsíce počínaje dnem instalace

Pravidelné platby

Měsíční paušál za dohodnutý počet výtisků	3 400,00 Kč
Výše pravidelného základního měsíčního nájmu	500,00 Kč

Objednané služby, splatné při uzavření smlouvy

Číslo služby	Název přiobjednané služby	Cena bez DPH
	Poplatek za uzavření smlouvy	100,00 Kč
	Poplatek za přípravu a instalaci stroje v rozsahu 1 hodin	200,00 Kč
	Poplatek za instalaci ovladačů na 3 PC	300,00 Kč
	Zaškolení obsluhy v rozsahu 1 hodin	400,00 Kč
	Dopravné při instalaci	500,00 Kč
		0,00 Kč
		0,00 Kč

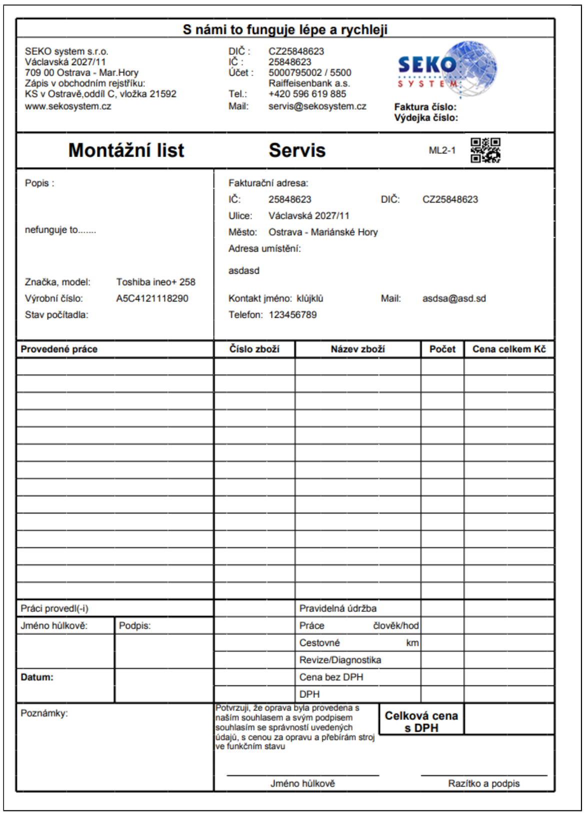
Zvláštní ujednání:
6. Místo instalace
Místo instalace: Ostrava

Tato smlouva se všemi jejími oddíly v bodech 1 až 6, jakožto i Všeobecné podmínky uvedené na druhé straně smlouvy a veškeré přílohy byly řádně přečteny. Na důkaz souhlasu se smlouvou a všemi jejími podmínkami připojují účastníci svoje vlastnoruční podpisy.

V Ostravě dne .... V...

Razítko a podpis pronajímatele *Razítko a podpis nájemce*

(a) Smlouva



S námi to funguje lépe a rychleji

SEKO system s.r.o.
Václavská 2027/11
709 00 Ostrava - Mar.Hory
Zápis v obchodním rejstříku:
KS v Ostravě,oddíl C, vložka 21592
www.sekosystem.cz

DIČ : CZ25848623
IČ : 25848623
Účet : 5000795002 / 5500
Raiffeisenbank a.s.
Tel.: +420 596 619 885
Mail: servis@sekosystem.cz

Faktura číslo:
Výdejka číslo:

Montážní list **Servis** ML2-1

Popis :
nefunguje to.......
Značka, model: Toshiba ineo+ 258
Výrobní číslo: A5C4121118290
Stav počítadla:

Fakturační adresa:
IČ: 25848623 DIČ: CZ25848623
Ulice: Václavská 2027/11
Město: Ostrava - Mariánské Hory
Adresa umístění:
asdasd
Kontakt jméno: klůjklů Mail: asdsa@asd.sd
Telefon: 123456789

Provedené práce	Číslo zboží	Název zboží	Počet	Cena celkem Kč

Práci provedl(-i)
Jméno hůlkově: Podpis:
Datum:

Položka	
Pravidelná údržba	
Práce	člověk/hod
Cestovné	km
Revize/Diagnostika	
Cena bez DPH	
DPH	

Poznámky:

Potvrzuji, že oprava byla provedena s naším souhlasem a svým podpisem souhlasím se správností uvedených údajů, s cenou za opravu a přebírám stroj ve funkčním stavu

Celková cena s DPH

Jméno hůlkově Razítko a podpis

(b) Montážní list

Obrázek 6: Exportované PDF soubory

4.10 Evidence strojů a příslušenství

V rámci evidence strojů, jsou 2 druhy strojů: přiřazené (smlouvě, firmě) nebo nepřiřazené. Toto dělení je důležité pro vyhledávání historie stroje. Každý stroj je buď propojený na šablonu a je tedy přesně definovaný typ a nebo ručně napsaný typ, aby nemusely být šablony na 100% použitých strojů. Při předání stroje od dodavatele (dopravce) se stroj vloží do systému a je v něm zapsaný, ikdyž není prodán a nebo umístěn u zákazníka. V tomto se liší od starého systému.

Jako hlavní identifikátor každého stroje se považuje sériové číslo, ale například některá příslušenství, sériové číslo nemají.

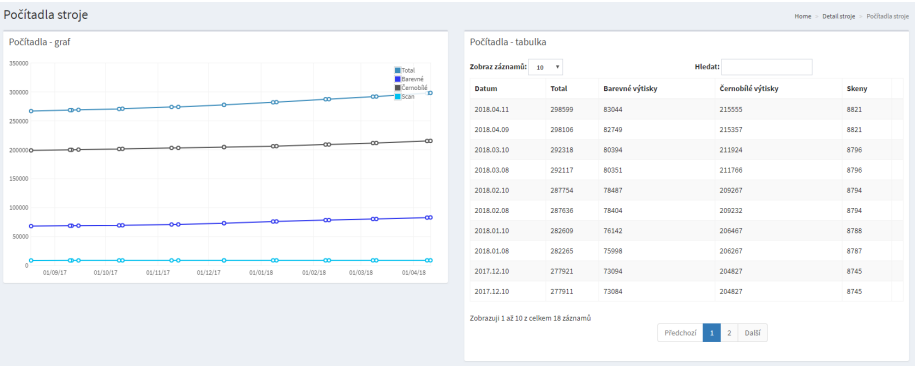
Stroje se mohou vyhledat podle firmy, kde jsou přiřazeny, podle sériového čísla a pak stroje, které nejsou nikde přiřazeny (jsou na skladě).

Když stroj přijde od dodavatele, zaměstnanec vloží stroj do sytému (buď se šablonou, nebo bez). Stroj je tedy samostatná jednotka, která bude v systému stále, i po likvidaci. Když se stroj prodá, vytvoří se balíček, který obsahuje základní informace, jako kontakt, umístění, atd. Ke každému balíčku se můžou přidělit příslušenství, které jsou namontovaná na stroji a tvoří tak nějaký celek.

U smlouvy se nejprve vytvoří smlouva, poté se smlouvě přiřadí typy a příslušenství z šablon (před podpisem není jasné, který konkrétní stroj - výrobní číslo - bude zákazníkovi nainstalován) a až po podpisu se podle přiřazených šablon vybere konkrétní stroj a příslušenství. V rámci dodatků se zákazníkovi může vyměnit stroj kus za kus (stejný model) nebo i za jiný model.

Detail stroje obsahuje základní informace. Zároveň je to takový rozcestník na další výpisy.

U detailu stroje jsem udělal graf počítadel, kde jsem hledal správnou knihovnu, aby graf mohl vypadat tak, jak bylo požadováno. V některých knihovnách byl problém mít nepravidelné periody přidávání záznamů. Jak lze vidět v grafu na obrázku níže, stroje většinou zasílají 2 počítadla měsíčně. Problém je, že je mohou posílat pouze, když je stroj zapnutý - proto se posílají 2, aby se snížila pravděpodobnost vypnutí. Také je možnost vyžádat poslání počítadla manuálně.



Obrázek 7: Detail stroje - graf počítadel

Strojová agenda je z většiny hotová, chybí jen několik výpisů a editačních formulářů.

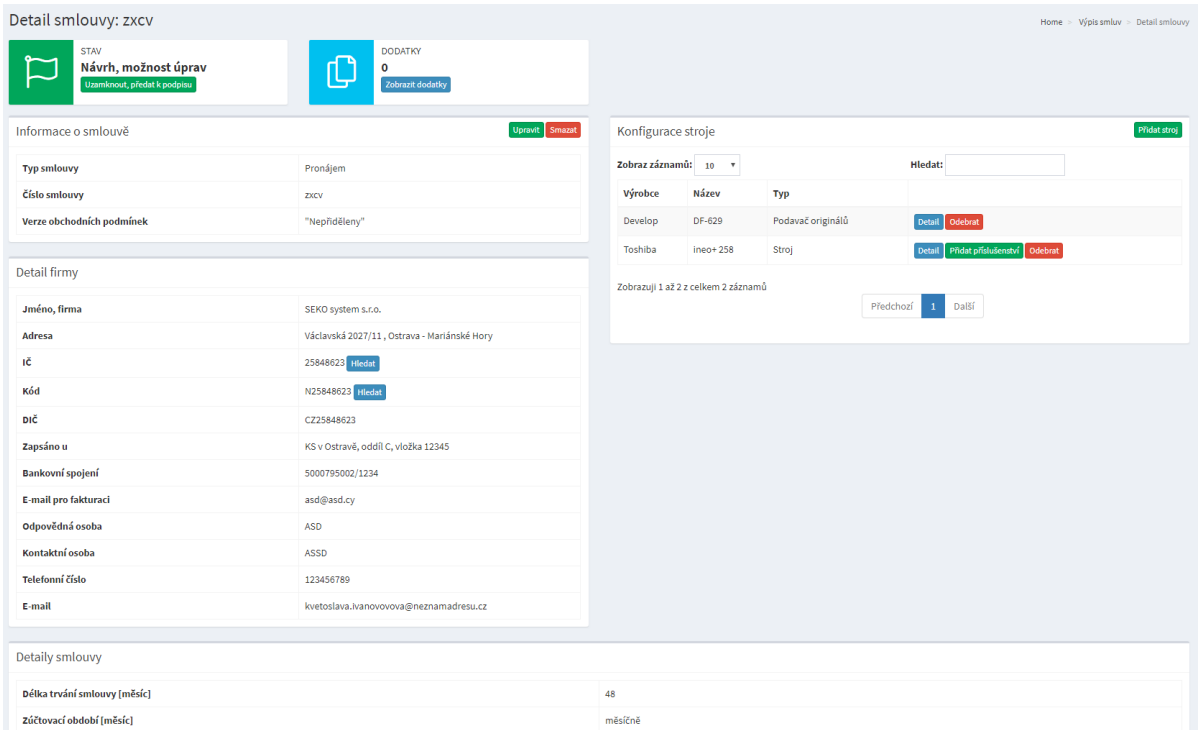
4.11 Systém smluv

V rámci smluv se neřeší jen evidence smluv samotných a jejich postup přidávání. Znamená to, že smlouva má 3 stádia: vytvořená smlouva, může se upravovat. Další stádium: čekání na podpis, tehdy se může vygenerovat PDF soubor i s všeobecnými obchodními podmínkami, které se řeší samostatně. Tento soubor se pak vytiskne a podepíše. Když se smlouva podepíše, smlouva přejde do posledního, třetího stavu: podepsáno. V rámci smluv jsou také řešeny dodatky, ale zároveň je kladen důraz na vyhledání historie všech dodatků.

Při vytváření smlouvy se ví pouze typ stroje a příslušenství. Po podepsání smlouvy se teprve přiřadí smlouvě konkrétní zařízení. První přiřazení je bez dodatku, každé další se řeší pomocí dodatku.

V dodatcích se řeší změna údajů ve smlouvě. Mají několik způsobů použití. Nejprve, aby se nemusely upravovat pro některé zákazníky všeobecné obchodní podmínky, tyto úpravy se řeší pomocí dodatků. Z pohledu aplikace se jedná nejen o úpravu zadaných údajů (např. ceny kopií, ceny poplatků, kontakt firmy, atd.), ale i o text, který se tak přiřadí smlouvě, kde se můžou upravovat např. všeobecné obchodní podmínky, aby nemusela existovat speciálně nová verze pro tuto smlouvu. Další využití je, že v průběhu platnosti smlouvy se upraví konfigurace stroje a nebo konkrétní zařízení.

Smlouvy lze vytvářet a pracovat s nimi, nicméně nejsou řešeny dodatky, které budu řešit později.



Obrázek 8: Detail smlouvy v režimu editace

4.12 Správa obchodních podmínek

Při generování smluv se vkládají podmínky, aby se pokaždé nemusely nahrávat. Ve správci podmínek, ať už všeobecných obchodních podmínek nebo podmínek pro software, jsem vytvořil nahrávací formulář s kontrolou pro PDF. V seznamu všech podmínek se vyberou jen jedny aktuální podmínky, které se vkládají do všech nových smluv. Když nejsou aktivované žádné podmínky, nová smlouva nejde vygenerovat. Obchodní podmínky se mění jen málokdy, proto je kladen důraz na dodatky, kde se řeší individuální změny, které vyžadují konkrétní zákazníci.

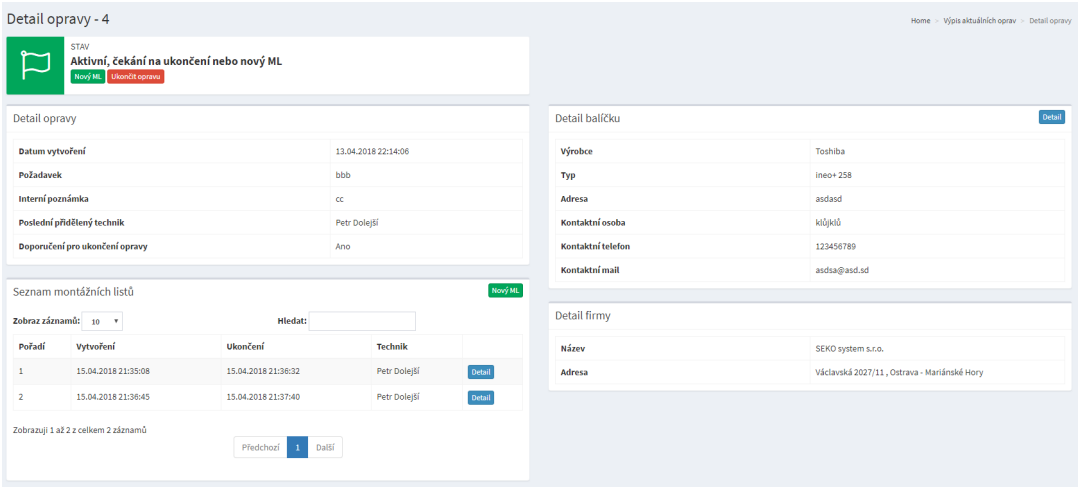
Podmínky lze nahrávat a spravovat bez žádných omezení.

4.13 Evidence oprav

Pro vytvořené balíčky lze vytvořit opravu. Někdo, nejčastěji sekretářka, která je u telefonu, vytvoří v systému opravu. Šéftechnik ji někomu přiřadí. V té chvíli se vygeneruje PDF soubor - montážní list. Výhoda, oproti starému systému je, že si ho může technik, pokud ho někde zapomene, vytisknout přímo u zákazníka. Opět jde o vkládání textu do buněk tabulky v Excelu a následný export do PDF. Po potvrzení od technika, že oprava je dokončená, technika zadá nějaké shrnutí do systému a doporučení šéf technikovi, zda je oprava již dokončená, nebo ne. Ten pak buď vytvoří nový montážní list, nebo opravu ukončí. Poté sekretářka zadá do systému údaje, např. cena práce a cestovné, ze kterých se budou vypočítávat mzdy. Než se ukončí první montážní list, oprava se může zrušit, aniž by se cokoliv započítávalo technikovi do mzdy.

Do budoucna se počítá, že na základě dokončených oprav systém bude počítat technikům mzdy, nebo alespoň bude vypočítávat hodnot, které zjednoduší práci. Opět je vyžadováno, aby byly všechny opravy dohledatelné, vůči všem součástem v balíčku stroje.

Opravy jsou celé dokončeny. V této části chybí jen napojení na výpočet mezd, které nebyly v první fázi projektu.



Obrázek 9: Detail opravy

5 Závěr

Vytvořil jsem část požadovaného informačního systému, celý jsem díky časové náročnosti nestihl, nicméně, již vytvořené části uživatelé testují a dávají mi zpětnou vazbu na úpravy a vylepšení.

V rámci této odborné praxe jsem využíval vědomostí, které jsem nabyl v průběhu studia. V průběhu praxe jsem využíval jazyky SQL, C# a JavaScript. Pro kódování HTML a CSS. Znalosti jsem nabyl v předmětech:

- Programovací jazyky (PJ2)
- Algoritmy (ALG1, ALG2)
- Úvod do databázových systémů (UDBS)
- Vývoj informačních systémů (VIS)
- Architektura .NET (AT .NET)
- Databázové a informační systémy (DAIS)
- Vývoj informačních aplikací (VIA)
- Správa Windows systémů (SWS)
- Tvorba aplikací pro mobilní zařízení (TAMZ1)

Poznal jsem kolektiv lidí, kteří ve firmě pracují ale také jsem z dřívější praxe v této firmě poznal práci technika. To mi pomohlo lépe pracovat s lidmi ve firmě i navrhovat procesy.

Ve vykonávání praxe mi scházely vědomosti v programování dynamických webových aplikací, které jsme to probírali okrajově na konci předmětu AT .NET. Pro mapování jsem mohl použít ORM, co jsme dělali v několika předmětech, ale neměl jsem ponětí o používání nějakých frameworků.

Celkově tuto zkušenost beru velmi pozitivně, protože jsem si zkusil pracovat samostatně, řešit problémy sám, ale na rozdíl od školních projektů jsem se musel držet toho, co od systému vyžadovala firma a dívat se na celý projekt tak, že to bude fungovat několik let a ne, že to jen na cvičení převedu cvičícímu.

S firmou jsem domluvený, že v programování toho IS budu pokračovat, nejen, než se to spustí, ale i na další rozšíření, které se plánují, včetně mobilní aplikace a rozšíření i pro obchodní oddělení.

Literatura

[1] *S22.Imap - knihovna* [Online]. [cit. 21. března 2018].
Dostupné z: `https://github.com/smiley22/S22.Imap`

[2] *ASP.NET MVC TUTORIÁLY* [Online]. [cit. 21. března 2018].
Dostupné z: `http://www.jiristepanek.cz/uhk/asp-net-mvc-tutorialy`

[3] *PDFsharp - knihovna* [Online]. [cit. 21. března 2018].
Dostupné z: `http://www.pdfsharp.net/`

[4] *GemBox - knihovna* [Online]. [cit. 21. března 2018].
Dostupné z: `http://www.gemboxsoftware.com/`

[5] *QRCoder - knihovna* [Online]. [cit. 21. března 2018].
Dostupné z: `https://github.com/codebude/QRCoder/`

[6] *AdminLTE - šablona* [Online]. [cit. 21. března 2018].
Dostupné z: `https://adminlte.io/`

[7] *Stack Overflow* [Online]. [cit. 21. března 2018].
Dostupné z: `https://stackoverflow.com/`

[8] *ITnetwork.cz* [Online]. [cit. 21. března 2018].
Dostupné z: `https://www.itnetwork.cz/`